On the Impact of IoT Traffic on the Cellular EPC

On the Impact of IoT Traffic on the Cellular EPC

Christian Vitale, Carla Fabiana Chiasserini, Francesco Malandrino
Politecnico di Torino, Italy
lastname@polito.it

***Abstract*—One of the most disruptive innovations in next-generation cellular networks will be the massive support of Machine Type and IoT (MTC/IoT) communications. This type of communications exhibits very different requirements from traditional cellular traffic: in MTC/IoT, the same base station may need to provide service to thousands of nodes, each of them transmitting small and infrequent data. In this context, it is critical to evaluate the impact of MTC/IoT on the Evolved Packet Core (EPC) network. We do so by quantifying analytically the signaling load on the EPC due to MTC/IoT bearer instantiation in both standard and 3GPP IoT-optimized LTE networks. Our analysis, validated via simulation, provides useful insights on the impact of the traffic load on each component of the EPC, as well as on the system design.**

I. Introduction

Smart cities, connected cars, wearables and industry 4.0 are only some of the examples that promote the use of Machine Type and IoT (MTC/IoT) communications in everyday life. The number of objects with sensing and communications capabilities are therefore envisioned to increase exponentially in the next few years, both by the academic community [1] and the general public [2]. While MTC/IoT communications, hereinafter referred to as IoT, have been typically relying on ad-hoc solutions in the recent past, large-scale deployments, as the ones required in the aforementioned examples, demand broad, reliable and efficient connectivity.

In this scenario, the cellular network emerges as a good candidate infrastructure for large-scale IoT systems [3], although they are very different from those supporting human-type communications. Most notably, a base station, instead of serving a handful of terminals, has to serve thousands of objects in IoT, each of them transmitting infrequent and small data. This implies that, for each data item an IoT source has to transmit, a control procedure may need to be executed for establishing an end-to-end connectivity, i.e., the so-called bearer instantiation. Such a control procedure involves the traffic sources, the base stations, and the cellular network, i.e., the Evolved Packet Core (EPC) network. Thus, the battery of the IoT sources, which typically has very little capacity, and the EPC may be strained by this enormous amount of control traffic. In order to address this shortcoming, 3GPP recently standardized some enhancements to the control procedures required by IoT traffic sources [4], including: the NB-IoT, a dedicated variant of LTE supporting only limited functionalities; the User-Plane C-IoT optimization, where the bearer for the IoT source is only suspended if the terminal does not send frequent data, instead of being released; and the Control-Plane C-IoT optimization, where end-to-end connectivity is granted to the IoTs at almost no cost.

The enhancements introduced by 3GPP allow reducing as much as possible the interaction of the IoT sources with the cellular infrastructure, hence reducing the control procedures overhead at the IoT source side. Nonetheless, the effect of such enhancements on the EPC components is still unclear. Our objective in this paper is to shed some light on the performance of the EPC network when standard or IoT enhanced control procedures are in place. This is particularly relevant in the context of the virtualization of the mobile core network, where the capacity of each component of the EPC can be tuned over time. How such tuning is performed depends on the control traffic load, thus the knowledge of the behavior of the EPC when handling thousands of IoT sources is essential to formulate scaling in/out algorithms of the EPC component capacity. More in detail, our main contributions are as follows:

- by adopting a realistic IoT traffic model, we analytically evaluate the average number of bearer instantiations per second that the EPC has to manage;
- we assess which entity of the EPC bears the highest burden;
- we compare the standard and enhanced procedures for the support of IoT traffic in cellular networks, and highlight the effectiveness of the enhancements that 3GPP standards recently introduced.

II. Related Work

Several works have dealt with IoT in the context of cellular networks, highlighting the shortcomings exhibited by control procedures. In particular, [5] has shown that an NFV-based solution to EPC implementation outperforms a solution exploiting SDN only, when a large number of IoT sources have to be served, i.e., under high control traffic conditions. The study in [6], instead, extends the standard EPC network by including new entities that are dedicated to IoT traffic. Although such an approach can improve the system performance, it implies several modifications to standard specifications.

Fewer works have aimed at evaluating the control traffic load that the EPC components have to bear if standard or enhanced bearer instantiations are in place. Among these, [7] proposes a mechanism for the aggregation of different IoT bearers and evaluates analytically its effectiveness over the standard procedure. The IoT traffic model adopted in [7], however, is very simple and 3GPP IoT optimized procedures are not considered. Similar observations hold for the study in

[8], which analitically models control procedures of video and IoT traffic on the EPC.

At last [9] proposes the IoT traffic model we adopt in this paper. The focus of [9], however, is on the simulation study of scenarios with millions IoT sources: neither an analytical evaluation of the IoT traffic nor a study of the EPC control procedures are presented.

III. System Model

In this paper, our purpose is to shed light on the performance of the EPC network when it has to handle the control traffic of thousands IoT sources. Thus, after a brief description of the notation adopted in the following sections (Sec. III-A), we introduce the main components we will refer to in our analysis, i.e., the EPC network (Sec. III-B), the bearer instantiation procedure (Sec. III-B1), and the model we adopt for IoT traffic (Sec. III-C).

A. Notation

We indicate with $\mathbb{P}(X)$ the probability of a specific event X. The probability mass function (pmf) of a discrete random variable X at x is denoted by $f_X(x) = \mathbb{P}(X = x)$. Finally, the evaluation at x of the pmf of X conditioned to the random variable Y, when $Y = y$, is denoted by $f_{X|Y}(x|y) = \mathbb{P}(X = x|Y = y)$.

B. EPC network

The EPC includes four main components [4], as depicted in Fig. 2:

- Serving Gateway (S-GW), mainly routes data traffic and acts as anchor point when the terminals perform handover between eNBs;
- PDN Gateway (P-GW), acts as ingress and egress point of the mobile network, and is responsible for policy enforcement;
- Mobility Management Entity (MME), is the termination point of the terminals control channels. The MME authenticates and tracks registered terminals and, most importantly, it handles bearer activation, i.e., the MME instantiates valid logical connections between the terminals and the P-GW;
- Home Subscriber Server (HSS), is a central database where terminal-related information are stored. The HSS assists the MME in the terminal authentication.

MME, P-GW and S-GW are different pieces of equipment, or, in the case of a virtual EPC, they typically run on different machines; the MME is then connected only to the S-GWs for bearer establishment and mobility management, while the P-GW handles the data traffic to/from several S-GWs. The MME mainly provides two important and recurrent control traffic procedures to ordinary terminals: bearer instantiation and handover. Nevertheless, 3GPP has introduced a new variant of LTE, namely, the NB-IoT, which does not support handover. In this case, at each change of eNB, an IoT source simply requests a new bearer instantiation when it has some traffic to send/receive. For this reason, in this paper we focus on the control traffic due to bearer instantiation by IoT sources, leaving the analysis of handover procedures to future investigation.

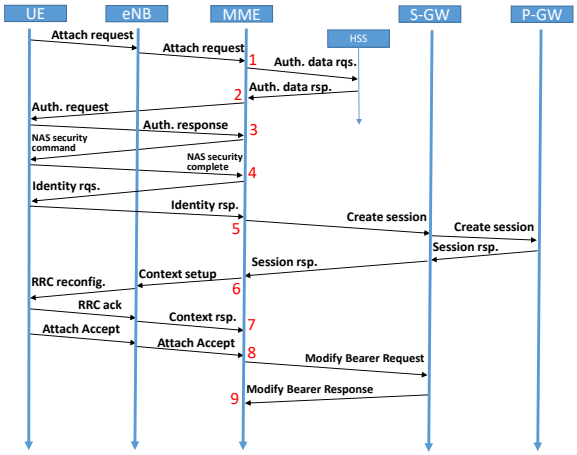


Fig. 1: Standard attach and bearer instantiation procedure.

1) Bearer instantion: In cellular networks a terminal can send or receive traffic only if there exists a logical connection between the terminal and the corresponding P-GW, i.e., only if the MME has completed the bearer instantiation procedure for that terminal. Whenever the time interval between subsequent packets of a terminal is larger than the Inactivity Timer of the eNB, the network resources associated with the terminal's bearer are released and the terminal gets detached from the network. In this case, before some new traffic can be transferred, the terminal has to re-attach to the cellular network and request a new bearer instantiation [8].

Fig. 1 details the attach/bearer instantiation procedure. In total, the MME has to perform six operations: attach, authentication, identity verification, session creation, context setup, and setting of the bearer options. As shown in the figure, if no previous terminal's context information is present in the network, the MME is required to process nine different control messages for each attach/bearer establishment.

The 3GPP enhancements introduced in the standard to favor MTC/IoT communications, namely, Attach without PDN connectivity, User Plane C-IoT optimization and Control Plane C-IoT optimization, aim at reducing the energy consumption of IoT sources by reducing the interaction between the IoT sources and the infrastructure during the control procedures. They indeed differ in the degree of interaction between the IoT sources and the infrastructure, as discussed below.

- Attach without PDN connectivity: at the expiration of the Inactivity Timer, the network resources associated with the bearer are released, but the terminal is not detached. When the traffic source has some new traffic to transmit, it has to establish a new bearer, but it does not have to re-attach, or to initiate a new session. Attach without PDN connectivity may be used for any type of terminal, and it is not specific to IoT sources;

- *User Plane C-IoT optimization*: at the expiration of the Inactivity Timer, the bearer of the IoT source is not released, but it is just suspended. When a new packet is transmitted from/to the IoT source, the bearer is quickly reactived;
- *Control Plane C-IoT optimization*: it allows the IoT source to transmit only small data. Instead of asking actively for a bearer instantiation, the IoT source piggy-bags its data in a control message towards the MME. The MME is responsible for the bearer instantiation, the integrity check of the IoT data, and the data forwarding towards the right S-GW. While this enhancement allows the IoT source to send its data using one single message (without any bearer instantiation request), it may represent a significant burden for the MME.

For lack of room, the handshakes of the three enhanced procedures are not shown here, but they can be found in [4].

At last, we remark that, although the above enhancements have been introduced mainly to reduce the IoT energy consumption, in this paper we focus on their impact on the EPC, not on the IoT sources.

C. IoT traffic model

As mentioned, whenever the time interval between subsequent packets of an IoT source is larger than the Inactivity Timer of the eNB, the bearer is released and a new bearer has to be re-established, if the IoT source has some new traffic to send/receive. Depending on the IoT traffic pattern, the time between subsequent packets may vary significantly and so does the number of bearer requests of an IoT source. In the following, we adopt the traffic model described in the 3GPP standard [10] for MTC. According to [10], traffic sources are organized in groups and, within each group, packet transmissions are quasi-synchronous. As an example, consider a group of sensors monitoring a geographical area and programmed to send an alarm to a server in the Internet when a specific event occurs. Upon an event occurrence, the sensors of the group that are closer to the event generate and transmit the alarm first, while the others react with some delay. In other words, a chain reaction triggers alarms from the sensors belonging to the same group, leading to a quasi-synchronous behavior. In particular, in [10] each group activity is divided in periods of duration T and, in each period, the quasi-synchronicity of the sensors described above, is captured by a beta(3,4) distribution representing the aggregate traffic generated within a group over time. The above system model is also depicted in Fig. 2.

Interestingly, the 3GPP model for the aggregate IoT traffic has been used in [9] to derive a model of traffic generation at each IoT source within a group. Specifically, [9] models an IoT source as a Markov Chain including two states, named *alarm* and *regular operation*, which we denote in the following by A and R, respectively. The period T of the IoT traffic pattern is divided into an arbitrary number of slots N, each of duration Δt, i.e., $N = \frac{T}{\Delta t}$. In state A, the generic IoT source sends packets following a Poisson distribution with

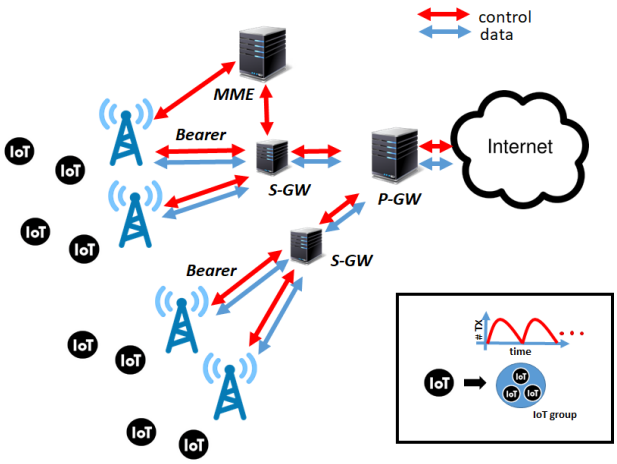


Fig. 2: Overview of the adopted system model.

mean $\lambda_1 = 1$ packet/slot. In state R, instead, the IoT source transmits packets following a Poisson distribution with mean $\lambda_2 = 0.0005\frac{\Delta t}{T}$ packet/slot, i.e., in state R an IoT source transmits on average 0.0005 packet/s; such traffic represents control packets at the application layer such as keep-alive procedures or synchronization packets. Both in state R and A, the packet size is set to $L = 100$ bytes [9].

At each time slot n, with $n = \{1, ..., N\}$, the IoT source may move from one state to the other. When in state A, the IoT source moves to R in the next slot with probability 1. On the contrary, the transition from R to A occurs with probability $f_s(n)$, which depends on the considered slot (s denotes the generic slot in which a transition occurs). To let the aggregate traffic generated by the IoT sources match the beta(3,4) shape, $f_s(n)$ is directly obtained from the sampling of such distribution [9]:

$$f_s(n) = \text{beta}(n\Delta t)\frac{\Delta t}{T}, \quad (1)$$

where beta$(n\Delta t)$ is the beta(3,4) shape sampled at time $n\Delta t$.

IV. Characterizing IoT control traffic

Here we draw on the single-source IoT traffic model presented in [9] and characterize the IoT control traffic load, in terms of:

- f_P, the pmf of the number of time slots between two subsequent visits to state A by an IoT source with at least one packet transmission. Such a quantity also corresponds to the number of slots between two consecutive forwarding operations by the MME towards the S-GW, related to data originated by the same IoT source, when the Control Plane C-IoT optimization is in place (Sec. IV-A);
- $\mathbb{E}[M]$, the average number of packets per second that the MME has to forward to the S-GW for a single IoT source, when the Control Plane C-IoT optimization is adopted (Sec. IV-A);
- $\mathbb{E}[B]$, the average number of bearer instantiations per second that each IoT source requests, when any of the

bearer establishment procedures introduced in Sec. IV-B is in place.

A. Computing f_P and $\mathbb{E}[M]$

We first consider the Control Plane C-IoT optimization, in which the MME acts as a relay for each packet transmitted/received by an IoT source. To compute f_P, we neglect the packets transmitted in state R – a fair assumption if T is sufficiently short since, on average, an IoT source transmits a packet every 2000 s while being in R. Under these conditions, f_P corresponds to the pmf of the number of slots between two alarms relayed by the MME for the same IoT source. We denote such a quantity by P; note that the P number of slots may span over several subsequent periods, each of duration T.

Let us now underline an important property of the IoT traffic model we adopted. Considering (1), an IoT source moves from state R to state A at a given time slot with a probability that does not depend on the past. Thus, the number of slots between two subsequent packet transmissions does not depend on the period in which the first packet was transmitted, but only on the particular slot within that period. We then compute f_P through the law of total probability considering N different cases, one for each slot in a period:

$$f_P(m) = \sum_{n=1}^{N} f_{P|\tau}(m|n) f_\tau(n), \tag{2}$$

where τ represents the slot of the period in which the IoT source last visited state A and transmitted a packet. From (2), we have to compute $f_{P|\tau}(m|n)$ and $f_\tau(n)$ in order to obtain $f_P(m)$. We start by computing $f_{P|\tau}(m|n)$:

$$f_{P|\tau}(m|n) = f_{\alpha|\tau}(m|n)(1-e^{-1}) + \sum_{k=1}^{\infty}\sum_{i=k}^{m-1} R_{k,n}(i) f_{\alpha|\tau}(m-i|\mathrm{mod}(n+i,N))(1-e^{-1}), \tag{3}$$

where (i) α is the time interval between two subsequent visits to A, (ii) $(1-e^{-1})$ is the probability that the tagged IoT source transmits at least one packet when visiting state A, and (iii) $R_{k,n}(i)$ represents the probability that an IoT source visits k times state A without transmitting any packet, the first time in slot n, the last one i slots after n. More specifically, (3) is computed as the sum of the probabilities of the following events:

- the IoT source visits state A only once, m slots after n, and it transmits at least one packet (first term of the right-hand side of (3));
- the IoT source visits $k+1$ times state A after transmitting a packet in slot n, but it transmits (at least) one packet only during its last visit to state A, m slots after n. As an example, for k=1, the source visits A twice: once i slots (with $i < m$) after n (without transmitting any packet), and once m slots after n (transmitting at least one packet). Indeed, the second term of the right-hand side of (3), for each intermediate i, is given by the product of the probability $R_{k,n}(i)$ and the probability that the new transition to state A happens in $m-i$ slots.

To complete the computation of $f_{P|\tau}(m|n)$, we need to derive $f_{\alpha|\tau}(m|n)$ and $R_{k,n}(i)$. Directly from (1), $f_{\alpha|\tau}(m|n)$ is computed as the probability of having: one transition from A to R (with probability 1 in slot $n+1$), $m-2$ slots where there is no transition, and one transition from R to A exactly m slots after n:

$$f_{\alpha|\tau}(m|n) = f_s(\mathrm{mod}(n+m,N)) \prod_{j=n+2}^{n+m-1} (1-f_s(\mathrm{mod}(j,N)) \ , \tag{4}$$

with $f_{\alpha|\tau}(1|n) = 0$.

$R_{k,n}(i)$ is instead computed recursively, accounting for the probability that after k visits to state A, an IoT source has not transmitted any packet yet. More in detail, $R_{1,n}(i)$ is given by:

$$R_{1,n}(i) = f_{\alpha|\tau}(i|n) e^{-1}, \tag{5}$$

i.e., as the probability that the IoT source visits A once, i slots after n, without transmitting any packet. $R_{k,n}(i)$ can be computed, through the law of total probability, as:

$$R_{k,n}(i) = \sum_{j=k-1}^{m-1} R_{k-1,n}(j) f_{\alpha|\tau}(i-j|\mathrm{mod}(n+j,N)) e^{-1}. \tag{6}$$

The summation on the right-hand side of (6) includes a term for each intermediate j ($R_{k-1,n}(j)$), where the probability of visiting state A for the $k-1$-th time without transmitting a packet, j slots after slot n, multiplies the probability that the new transition to state A happens in $i-j$ slots.

Finally, we compute $f_{P|\tau}(m|n)$ by leveraging on the result provided in [9], i.e.,

$$f_\tau(n) = f_s(n). \tag{7}$$

The intuition behind (7) is that $f_s(n)$ is strictly linked with the probability that the IoT source transmits a packet (see (1)).

From f_P we also obtain $\mathbb{E}[M]$, which can be obtained as the ratio between (i) the average number of packets transmitted by the tagged IoT source while in A, conditioned to the fact that the IoT source transmits at least one packet, and (ii) the average interval of time between two subsequent visits to A with at least one packet transmission by the IoT source, i.e., $\mathbb{E}[P]\Delta t$:

$$\mathbb{E}[M] = \frac{\sum_{k=1}^{\infty} \exp^{-1}/(k-1)!}{(1-\exp^{-1})} \cdot \frac{1}{\mathbb{E}[P]\Delta t}, \tag{8}$$

where the first term on the right hand side is directly obtained from the Poisson pmf with $\lambda_1 = 1$.

B. Computing $\mathbb{E}[B]$

We now focus on the Inactivity Timer T_I expiration. Whenever such an event occurs, a new packet transmission by an IoT source triggers a new bearer-related procedure. Specifically, the standard case requires a re-attach and bearer establishment, the Attach without PDN connectivity implies a new bearer establishment, the User Plane C-IoT optimization just requires

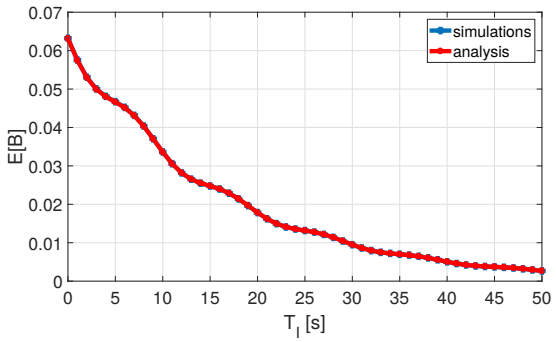


Fig. 3: $\mathbb{E}[B]$ vs. T_I: Comparison between analytical and simulation results.

a bearer resume procedure since in this case the bearer is just suspended, while the Control Plane C-IoT optimization demands directly the MME to initiate a bearer establishment procedure. Although the control traffic load induced by the timer expiration varies depending on the procedure that is in place, in all cases it is essential to compute the average number of times per second the timer expires, i.e., $\mathbb{E}[B]$.

Given the definition of P in Sec. IV-A, we compute $\mathbb{E}[B]$ as the ratio of the probability that a packet triggers a bearer request ($\mathbb{P}(P\Delta t > T_I)$), to the average time between two consecutive packet transmissions by the same IoT source ($\mathbb{E}[P]\Delta t$), i.e.,

$$\mathbb{E}[B] = \frac{\mathbb{P}(P\Delta t > T_I)}{\mathbb{E}[P]\Delta t}. \quad (9)$$

where $\mathbb{P}(P\Delta t > T_I)$ can be directly computed from f_P.

V. Model Validation and Exploitation

In this section we first validate the analytical results obtained in Sec. IV, i.e., $\mathbb{E}[M]$ and $\mathbb{E}[B]$, against simulations (Sec. V-A). Then we exploit our model to derive interesting comparisons among the standard and IoT enhanced bearer instantiation procedures (Sec. V-B).

A. Model Validation

To validate our model, we developed a Matlab simulator that implements the traffic model described in Sec. III-C. The parameters we used are as follows: $T = 10\,\text{s}$ (as specified in [4]), $\Delta t = 10\,\text{ms}$, and T_I ranging between 0 and 50 s. We performed extended experiments and compared $\mathbb{E}[M]$ and $\mathbb{E}[B]$ obtained analytically and via simulation.

The results obtained for $\mathbb{E}[B]$ are shown in Fig. 3, as T_I varies. As expected, the number of bearers per second an IoT source requests, decreases monotonically with increasing T_I. However, we recall that, the larger the T_I, the more context information the EPC has to store. More importantly, Fig. 3 highlights that the curve obtained via simulation and the one obtained analytically perfectly match, thus showing that our analysis perfectly captures the number of bearers per second each IoT source requests to the EPC.

For sake of brevity, we do not show here the results for $\mathbb{E}[M]$ but we mention that, for this metric, the difference between analysis and simulation is less than 0.04% in the worst case.

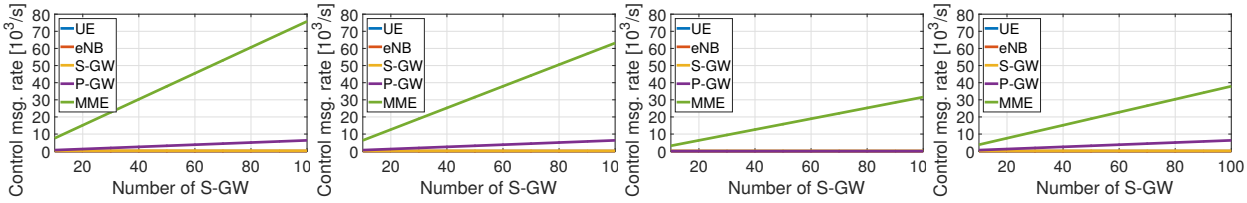
B. Comparison among bearer instantiation procedures

We now characterize the average load that each component of the EPC has to handle in the bearer instantiation procedures presented above. First, in Table I we present the number of messages that each component has to dispatch for each instantiated bearer, as per the 3GPP standard specifications [4]. In the table, we do not account for the initial RRC connection establishment, since it is a mandatory procedure involving only the IoT source and the eNB, as per classical data forwarding. The number of messages each component has to handle, however, includes those required to release (or suspend, in case of an User-Plane C-IoT optimization) a bearer, since, for each bearer established, a bearer is also released. As far as the Control-Plane C-IoT optimization is concerned, we account for two different operations: (i) bearer establishment/release and (ii) MME data forwarding. In the latter operation, the eNB piggybags the packet received from the IoT source in an S1-AP initial message, and the MME checks the integrity and decrypts the packet before forwarding it to the correct S-GW. Similarly, the IoT source has to simply piggybag its data packets in the messages transmitted during the RRC establishment procedure, without the need for any additional control message.

To characterize the traffic load that each component has to handle due to bearer-related procedures, we consider a scenario with 10 eNBs for each S-GW, each eNB covering 100 IoT sources, 1 MME, and 1 P-GW. The number of S-GWs in the system instead varies from 10 to 100, which allows us to analyse scenarios with different control load. As suggested in [11], the capacity of the different components of the EPC can be expressed in terms of messages/s. Fig. 4 depicts the control load that each entity has to handle in each scenario, in terms of the control message rate. For what concern S-GW, eNB, and UE, the quantities refer to a single entity and not to

TABLE I: Number of messages that each EPC component has to process for every bearer instantiation

Procedure	♯ Msg. IoT	♯ Msg. eNB	♯ Msg. MME	♯ Msg S-GW	♯ Msg. P-GW
Standard Bearer Inst.	6	6	12	4	1
Attach w/o PDN Con.	5	5	10	3	1
User-Plane C-IoT Opt.	2	6	5	2	0
Control-Plane C-IoT Bearer	0	2	5	3	1
Control-Plane C-IoT Data	0	1	1+Decrypt. +Integrity	0	0



(a) Standard procedure (b) Attach w/o PDN connectivity (c) User-Plane C-IoT opt. (d) Control-Plane C-IoT opt.

Fig. 4: Number of messages that each EPC component has to dispatch per second, as the number of S-GWs varies and with $T_I = 0$.

the whole set. As an example, the total message rate an MME has to handle under the Control-Plane C-IoT optimization, is:

$$\mathbb{E}[M_{CP}] = \mathbb{E}[B] \cdot 5Q + \mathbb{E}[M] \cdot DQ + \frac{Q}{\mathbb{E}[P]}, \qquad (10)$$

where (i) Q is the total number of IoT sources under a specific MME, (ii) 5 is the number of messages that the MME has to handle for each bearer establishment/release in the Control-Plane C-IoT optimization, (iii) $\frac{1}{\mathbb{E}[P]}$ is the message rate that the MME has to handle for each IoT source due to control traffic forwarding, and (iv) D is the integrity check and decryption load that the MME has to handle for each packet relayed for an IoT source, expressed in terms in number of messages. Computing D may not be trivial. Nevertheless, studies on commodity processors show that nowadays a packet of size L requires few hundreds of floating-point operations for encryption/decryption [12]; similar results are obtained also for packet integrity check. Since a single control message requires (roughly) one million floating-point operations [8], we neglected the load due to packet integrity check and decryption operations.

Fig. 4 further highlights that, in all bearer procedures, the MME is the one that has to dispatch the highest number of messages. In the best case (User-Plane C-IoT optimization), the MME still dispatches seven times more messages than any other entity. Finally, we conclude that, under the Control Plane C-IoT optimization, the interaction with the IoT sources is reduced to the minimum, but the control load on the MME remains comparable to the one observed for the other bearer establishment procedures.

VI. Conclusions

We considered the case where a cellular network supports massive IoT communications and analytically derived the consequent control traffic load on the EPC. Specifically, assuming a 3GPP-based model for IoT traffic sources, we chracterized the distribution of the time interval between two consecutive transmissions by the same source. Through this expression, we computed the control traffic that the EPC has to handle due to the support of IoT sources, when different connectivity procedures are in place, namely, the standard procedure, the Attach without PDN connectivity, the User Plane C-IoT optimization, and the Control Plane C-IoT optimization. Our analysis was validated through simulation results and provided some novel insights on the impact of massive IoT traffic on the EPC components, as well as on effectiveness of the different connectivity modes foreseen by the 3GPP standard.

VII. Acknowledgment

This work was supported by the European Commission through the H2020 5G-TRANSFORMER project (Project ID 761536).

References

[1] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green IoT: An investigation on energy saving practices for 2020 and beyond," *IEEE Access*, vol. 5, pp. 15 667–15 681, 2017.

[2] "Forbes prediction on iot," https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#3768bc2a1480, accessed: 2018-19-04.

[3] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120–128, 2017.

[4] 3rd Generation Partnership Project, "3GPP specification: 23.401; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," Tech. Rep., 2017. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/23_series/23.401/

[5] A. Jain, N. Sadagopan, S. K. Lohani, and M. Vutukuru, "A comparison of sdn and nfv for re-designing the lte packet core," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on*, 2016, pp. 74–80.

[6] V. Nagendra, H. Sharma, A. Chakraborty, and S. R. Das, "Lte-xtend: scalable support of m2m devices in cellular packet core," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 43–48.

[7] S. Abe, G. Hasegawa, and M. Murata, "Effects of C/U plane separation and bearer aggregation in mobile core network," *IEEE Transactions on Network and Service Management*, 2018.

[8] J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado, and J. M. Lopez-Soler, "Modeling and dimensioning of a virtualized mme for 5g mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4383–4395, 2017.

[9] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," in *Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on*, 2013, pp. 1–5.

[10] 3rd Generation Partnership Project, "3GPP specification: 37.868; RAN Improvements for Machine-type Communications," Tech. Rep., 2014. [Online]. Available: http://www.3gpp.org/ftp//Specs/archive/37_series/37.868/

[11] G. Hasegawa and M. Murata, "Joint bearer aggregation and control-data plane separation in lte epc for increasing m2m communication capacity," in *Global Communications Conference (GLOBECOM), 2015 IEEE*, 2015, pp. 1–6.

[12] "Encryption performance on commodity processors," https://calomel.org/aesni_ssl_performance.html, accessed: 2018-19-04.